IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application Serial No. 10/087,672
Filing Date 02/27/2002
Inventorship ..... Aascheim et al.
Applicant ..... Microsoft Corp.
Group Art Unit 2186
Examiner ..... Patel, Hetul B.
Attorney's Docket No. MS183343.01
Title: Open Architecture Flash Driver

APPEAL BRIEF

To: Commissioner for Patents
PO Box 1450
Alexandria, Virginia 22313-1450

From: Lance R. Sadler (Tel. 509-755-7262; Fax 509-755-7252)
Sadler, Breen, Morasch & Colby
422 W. Riverside Avenue, Suite 424
Spokane, WA 99201

Pursuant to 37 C.F.R. §41.37, Appellant hereby submits an appeal brief for application 10/087,672, filed February 27th, 2002. Accordingly, Appellant appeals to the Board of Patent Appeals and Interferences seeking review of the Examiner's rejections.

(1) Real Party in Interest

The real party in interest is Microsoft Corporation, the assignee of all right, title and interest in and to the subject invention.

(2) Related Appeals and Interferences

Appellant is not aware of any other appeals, interferences, or judicial proceedings which will directly affect, be directly affected by, or otherwise have a bearing on the Board's decision to this pending appeal.

(3) Status of Claims

Claims 1, 3-11, 13-19, 21-26, 28-34, and 36-44 stand rejected and are pending in this Application. The rejections of Claims 1, 3-11, 13-19, 21-26, 28-34, and 36-44 are appealed. Claims 3-5, 13-14, 17-19, 21, 22, 24-26, 31-32, 34, and 36-40 are original and hence bear the designator "(Original)". Claims 1, 6-11, 15, 16, 23, 33, and 41-44 are previously presented and hence bear the designator "(Previously presented)". Claims 2, 12, 20, 27, 35, and 35-52 have been canceled.

Claims 1, 3-11, 13-19, 21-26, 28-34, and 36-44 are set forth in the Appendix of Appealed Claims on page 39.

(4) Status of Amendments

The Final Office Action, which is the subject of this Appeal, was mailed November 20th, 2007 (herein the "Final Office Action").

No amendments were made to the claims subsequent to the final rejection.

(5) Summary of Claimed Subject Matter

A concise explanation of each of the independent claims is included in this Summary section, including specific reference characters. These specific reference characters are examples of particular elements of the drawings for certain claimed embodiments. It is to be appreciated and understood that the

claims are not to be limited to solely the elements corresponding to these reference characters and that this section is provided to comply with the requirement of 37 CFR § 41.37(c)(1)(v).

Claim 1 recites one or more computer storage media (Fig. 3 (304), Page 31 (line 18) through Page 32 (line 7)) comprising a flash memory driver that is executable by a computer to interface between a file system and one or more flash memory media (Fig. 3 (306), Page 2 (lines 6-17)), the flash memory driver comprising: flash abstraction logic that is invokable by the file system to manage flash memory operations without regard to the type of the one or more flash memory media (Fig. 3 (308), Fig. 4, Page 10 (line 4) through Page 11 (line 7)); and flash media logic configured to interact with different types of the flash memory media, wherein the flash media logic is programmable to permit users to match particular medium requirements of a specific manufacturer (Fig. 3 (310), Fig. 5, Page 11 (line 8) through Page 12 (line 3)); wherein the flash abstraction logic invokes the flash media logic to perform memory operations that are potentially performed in different ways depending on the type of the flash memory media (Fig. 3 (306-310), Page 2 (lines 13-17)), and further wherein the flash memory driver is flash memory medium agnostic (Fig. 3 (306), Page 9 (lines 4-8)), and wherein one of the flash memory operations includes performing wear-leveling operations (Fig. 13, Fig. 14, Page 8 (lines 22-25)) associated with the flash memory medium by way of circular and continuous advancement of a write pointer (Fig. 13 (302), Fig. 14, Page 22 (lines 14-22), Page 25 (lines 1-11)) and wherein the flash memory driver resides as a component within an operating system of the computer (Fig. 3 (304-310), Page 9 (lines 9-16)).

Claim 9 recites a flash driver embodied on a computer-readable storage medium (Fig. 3 (304-306), Page 2 (lines 6-17), Page 31 (line 18) through Page 32 (line 7)), comprising: flash abstraction logic, interposed between a file system and a flash memory medium (Fig. 3 (308), Page 2 (lies 10-13)), configured to: (a) map a logical sector status from the file system to a physical sector status of the flash

memory medium (Fig. 4 (404), Page 10 (lines 10-15), Page 11 (lines 1-3)); and (b) maintain memory requirements associated with operating the flash memory medium (Fig. 3 (308), Fig. 4 (402), Page 10 (lines 4-7)); wherein the flash driver is located remote from the flash memory medium (Fig. 3 (306, 100/200), Page 2 (lines 6-8), Page 9 (lines 9-16)), and wherein the memory requirements include managing wear-leveling operations (Fig. 13, Fig. 14, Page 8 (lines 22-25)) associated with the flash memory medium by way of circular and continuous advancement of a write pointer (Fig. 13 (302), Fig. 14, Page 22 (lines 14-22)), Page 25 (lines 1-11)), and wherein the flash driver resides as a component within an operating system of a computer (Fig. 3 (304-310), Page 9 (lines 9-16)); flash medium logic that is programmable to permit users to match particular medium requirements of a specific manufacturer (Fig. 3 (310), Fig. 5, Page 11 (line 8) through Page 12 (line 3)).

Claim 16 recites a flash driver embodied on a computer-readable storage medium (Fig. 3 (304-306), Page 2 (lines 6-17)), Page 31 (line 18) through Page 32 (line 7)), comprising: user programmable flash medium logic (Fig. 3 (310), Page 9 (lines 17-18)), configured to read, write and erase data to and from a flash memory medium, wherein the flash medium logic is programmable to permit users to match particular flash medium requirements of a specific manufacturer (Fig. 3 (310), Fig. 5, Page 11 (lines 11-17)); and flash abstraction logic, interposed between a file system and flash memory medium (Fig. 3 (308), Page 2 (lines 10-13)) to maintain universal requirements for the operation of the flash memory medium (Fig. 3 (308), Page 10 (lines 4-15)); wherein the flash memory driver is flash memory medium agnostic (Fig. 3 (306), Page 9 (lines 4-8)), and wherein the universal requirements include managing wear-leveling operations (Fig. 13, Fig. 14, Page 8 (lines 22-25)) associated with the flash memory medium by way of circular and continuous advancement of a write pointer (Fig. 13 (302), Fig. 14, Page 22 (lines 14-22)), Page 25 (lines 1-11)), and wherein the flash driver is

defined as a component within an application (Fig. 3 (304-310), Page 9 (lines 9-16)).

Claim 23 recites a processing device that uses a flash memory medium for storage of data (Fig 3 (300), Page 2 (lines 6-17)), comprising: a file system, configured to control data storage for the processing device (Fig. 3 (305), Page 8 (lines 10-17)); flash media logic (Fig. 3 (310), Page 9 (lines 17-18)), configured to perform physical sector operations to a flash memory medium based on physical sector commands (Fig. 3 (310), Fig. 5, Page 11 (lines 8-17)), wherein the flash medium logic comprises a set of programmable entry points that can be implemented by a user to interface with any type of flash memory medium selected (Fig. 5 (502), Page 11 (lines 19-25)) wherein the flash media logic is programmable to permit users to match particular flash medium requirements of a specific manufacturer (Fig. 3 (310), Fig. 5, Page 11 (line 8) through Page 12 (line 3)); and flash abstraction logic, configured to maintain flash memory requirements that are necessary to operate the flash memory medium (Fig. 3 (308), Page 2 (lines 10-13)), wherein the flash memory requirements include managing wear-leveling operations (Fig. 13, Fig. 14, Page 8 (lines 22-25)) associated with the flash memory medium by way of circular and continuous advancement of a write pointer (Fig. 13 (302), Fig. 14, Page 22 (lines 14-22), Page 25 (lines 1-11)), wherein the flash media logic and the flash abstraction logic comprise a flash driver (Fig. 3 (306-31), Page 2 (lines 10-17)).

Claim 33 recites in a processing device that uses a flash memory medium for storage of data, a method for driving the flash memory medium (Fig 3 (300), Page 2 (lines 6-17)), comprising: managing rules associated with operating the flash memory medium in a flash abstraction logic (Fig. 3 (308), Page 10 (lines 4-15)), wherein the flash abstraction logic comprises part of a flash driver (Fig. 3 (306, 308), Page 2 (lines 10-13); and issuing physical sector commands directly to the flash memory medium from a flash medium logic (Fig. 3 (310), Fig. 5, Page 11 (lines 8-17)), wherein the flash medium logic is programmable to permit users

to match particular flash medium requirements of a specific manufacturer (Fig. 3 (310), Fig. 5, Page 11 (line 8) through Page 12 (line 3)), and wherein the flash medium logic comprises part of the flash driver (Fig. 3 (306, 310), Page 2 (lines 13-14)); wherein the method is flash memory medium agnostic (Fig. 3 (306), Page 9 (lines 4-8)), and wherein one of the rules includes managing wear-leveling operations (Fig. 13, Fig. 14, Page 8 (lines 22-25)) associated with the flash memory medium by way of circular and continuous advancement of a write pointer (Fig. 13 (302), Fig. 14, Page 22 (lines 14-22), Page 25 (lines 1-11)), and wherein the method is performed by way of a component residing within an operating system of the processing device (Fig. 3 (304-310), Page 9 (lines 9-16)).

Claim 42 recites a computer storage media (Fig. 3 (304), Page 31 (line 18) through Page 32 (line 7) for a flash driver (Fig. 3 (306), Page 2 (lines 6-17)), comprising computer-executable instructions that, when executed, direct the flash driver to provide an interface between a file system, selected from one of a plurality of different file systems (Fig. 3 (305, 306), Page 9 (lines 1-4)), and a flash memory medium, selected from one of a plurality of different flash memory media (Fig. 3 (306, 100/200), Page 9 (lines 4-8)) wherein the flash driver is located as a component within an operating system and is remote from the flash memory medium (Fig. 3 (304-310), Page 9 (lines 9-16)), and wherein wear-leveling (Fig. 13, Fig. 14, Page 8 (lines 22-25)) of the flash memory medium is performed by way of circular and continuous advancement of a write pointer (Fig. 13 (302), Fig. 14, Page 22 (lines 14-22), Page 25 (lines 1-11)), and wherein the flash driver comprises programmable flash medium logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer (Fig. 3 (310), Fig. 5, Page 11 (line 8) through Page 12 (line 3)).

Claim 43 recites a computer storage media (Fig. 3 (304), Page 31 (line 18) through Page 32 (line 7)) for a flash driver (Fig. 3 (306), Page 2 (lines 6-17)), comprising computer-executable instructions that, when executed, direct the flash driver to: provide an interface between a file system, selected from one of a

plurality of different files systems (Fig. 3 (305, 306), Page 9 (lines 1-4)), and a flash memory medium, selected from one of a plurality of different flash memory media (Fig. 3 (306, 100/200), Page 9 (lines 4-8)); and manage a set of characteristics that are common to the plurality of different flash memory media at a flash abstraction logic (Fig. 3 (308), Page 10 (lines 4-15)); wherein the flash driver is flash memory medium agnostic (Fig. 3 (306), Page 9 (lines 4-8)), and wherein wear-leveling (Fig. 13, Fig. 14, Page 8 (lines 22-25)) of the flash memory medium is performed by way of circular and continuous advancement of a write pointer (Fig. 13 (302), Fig. 14, Page 22 (lines 14-22), Page 25 (lines 1-11)), and wherein the flash driver resides as a component within an operating system (Fig. 3 (304-310), Page 9 (lines 9-16)), wherein the instructions provide programmable flash medium logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer (Fig. 3 (310), Fig. 5, Page 11 (line 8) through Page 12 (line 3)), wherein the flash abstraction logic and the flash medium logic comprise the flash driver (Fig. 3 (306-31), Page 2 (lines 10-17)).

Claim 44 recites a computer storage media (Fig. 3 (304), Page 31 (line 18) through Page 32 (line 7)) for a flash driver (Fig. 3 (306), Page 2 (lines 6-17)), comprising computer-executable instructions that, when executed, direct the flash driver to: provide an interface between a file system, selected from one of a plurality of different files systems (Fig. 3 (305, 306), Page 9 (lines 1-4)), and a flash memory medium, selected from one of a plurality of different flash memory media (Fig. 3 (306, 100/200), Page 9 (lines 4-8));manage a set of characteristics that are common to the plurality of different flash memory media at a flash abstraction logic (Fig. 3 (308), Page 10 (lines 4-15)); and provide programmable entry points that can be optionally selected by a user to interface with the type of flash memory medium selected (Fig. 5 (502), Page 11 (lines 19-25)); wherein the flash driver is located as a component within an operating system (Fig. 3 (304-310), Page 9 (lines 9-16)) and is remote from the flash memory medium and the flash driver is flash memory medium agnostic (Fig. 3 (306), Page 9 (lines 4-8)),

and wherein wear-leveling (Fig. 13, Fig. 14, Page 8 (lines 22-25)) of a flash memory medium is performed by way of circular and continuous advancement of a write pointer (Fig. 13 (302), Fig. 14, Page 22 (lines 14-22), Page 25 (lines 1-11)), wherein the instructions provide flash medium logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer (Fig. 3 (310), Fig. 5, Page 11 (line 8) through Page 12 (line 3)), wherein the flash abstraction logic and the flash medium logic comprise the flash driver (Fig. 3 (306-31), Page 2 (lines 10-17)).

(6) Grounds of Rejection to be Reviewed on Appeal

The drawings stand objected to under 37 CFR 1.83(a).

Claims 23-26, 28-34, 36-41, and 43-44 stand rejected under 35 U.S.C. § 112, second paragraph, as failing to set forth the subject matter which applicant(s) regard as their invention.

Claims 1, 5-11, 15-18, 22-25, 29-33 and 37-44 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,799,168 to Ban ("Ban") in view of U.S. Patent No. 6,725,321 to Sinclair et al. ("Sinclair") further in view of U.S. Patent No. 5,875,478 to Blumenau ("Blumenau"), and U.S. Patent No.6,253,281 to Hall ("Hall").

Claims 3, 4, 13, 14, 19, 21, 26, 28, 34, and 36 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Ban in view of Sinclair and in view of Blumenau, Hall and U.S. Patent No. 6,493,807 to Martwick ("Martwick").

Claims 1, 5-11, 15-18, 22-25, 29-33, and 37-44 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Ban in view of Sinclair, in view of Hall and further in view of Blumenau.

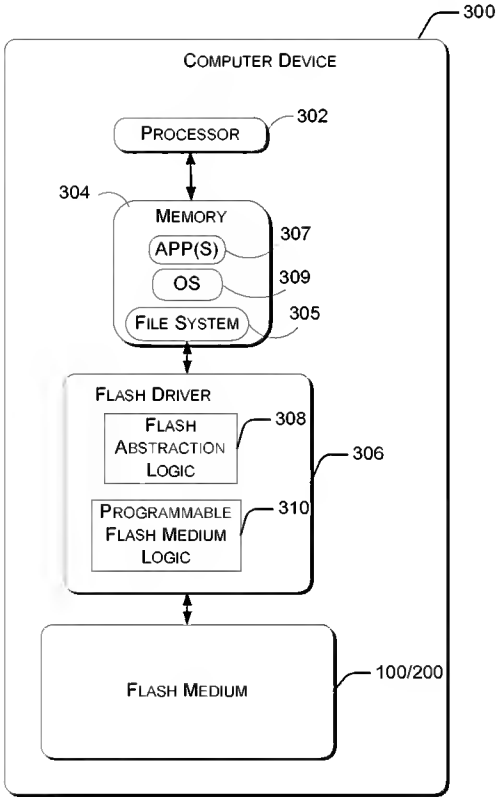
Claims 3, 4, 13, 14, 19, 21, 26, 28, 34, and 36 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Ban in view of Sinclair, further in view of Hall, further in view of Blumenau and further in view of Martwick.

(7) Argument

A. The objection to the drawings under 37 CFR 1.83(a) should be withdrawn

The Office has objected to Fig. 3 as failing to show every feature of the specified claims. Claims 1 and 16 recite: "wherein the flash memory driver resides as a component within an operating system of the computer". The Office argues that Fig. 3 does not depict the flash memory driver as residing within the operating system of the computer. For convenience, Fig. 3 is reproduced below:

Fig. 3 of Appellant's Disclosure



Page 9, lines 9-16 of Appellant's disclosure states:

> In the exemplary implementation, ***flash driver 306 resides as a component within operating system 309***, that when executed serves as a logical interface module between the file system 305 and flash medium 100/200. ***The flash driver 306 is illustrated as a separate box 306 for purposes of demonstrating that the flash driver when implemented serves as an interface***. Nevertheless, flash driver 306 can reside in other applications, part of the file system 305 or independently as separate code on a computer-readable medium that executes in conjunction with a hardware/firmware device.

The excerpt above clearly states that the "flash driver resides as a component within operating system 309" and that the "flash driver 306 is illustrated as a separate box 306 for ***purposes of demonstrating that the flash driver when implemented serves as an*** interface." Accordingly, Appellant's disclosure makes it clear that the flash driver is depicted in Fig. 3 as a separate box merely to show that driver can serve as an interface – but that the flash driver can reside within the operating system. Therefore, Appellant respectfully submits that claims 1 and 16 are supported by Fig. 3 and Applicant's disclosure. As such, this objection should be withdrawn.

B. The §112 rejection of claims 23-26, 28-34, 36-41, and 43-44 should be withdrawn

The Office has rejected claims 23-26, 28-34, 36-41, and 43-44 for reciting that the flash media logic and the flash abstraction ***comprise*** the flash memory driver. The Office argues that this element is not supported by Appellant's specification which teaches the flash memory driver ***comprises*** flash media logic and flash abstraction logic. In other words, the Office argues that claims 23, 33, 43, and 44 teach that B and C comprise A, whereas Appellant's specification teach that A comprises B and C.

"Comprise" can be defined as "***to include***" or "***compose***." (*see* www.merriam-webster.com/dictionary/comprise.) Therefore, Appellant respectfully submits that stating that B and C comprise (***compose***) A (B + C = A) is the same as stating that A comprises (***includes***) B and C (A = B + C), therefore stating that the flash media logic and flash abstraction logic comprise the flash memory driver is the same as stating that the flash memory driver comprises flash media logic and flash abstraction logic. Thus, Appellant respectfully submits that the rejection of claims 23-26, 28-34, 36-41, and 43-44 under 35 USC §112 should be withdrawn.

C. The rejections under 35 U.S.C. § 103(a) should be withdrawn because the Office has failed to make out a *prima facie* case of obviousness.

Claims 1 and 3-8

Claim 1

Claim 1 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Ban, Sinclair, Blumenau, and Hall. Appellant respectfully submits that the Office has failed to establish a *prima facie* case of obviousness for rejecting Claim 1 in the Final Office Action. The Office has failed to do so first by failing to establish that Ban teaches a flash memory driver comprising flash abstraction logic and flash media logic. Second, the Office improperly argues that Ban could be modified to teach the flash memory driver, as claimed, because Ban teaches directly away from the modification suggested by the Office. Third, the Office fails to establish that Hall, as used as an alternative to Ban, teaches a flash memory driver comprising flash abstraction logic and flash media logic. Fourth, the Office has used hindsight reconstruction to make out the rejection of this claim.

For the reader's convenience, the subject matter of Claim 1 is provided below, after which Appellant submits its arguments for Claim 1.

Claim 1 recites one or more computer storage media comprising a flash memory driver that is executable by a computer to interface between a file system and one or more flash memory media, the flash memory driver comprising:

- flash abstraction logic that is invokable by the file system to manage flash memory operations without regard to the type of the one or more flash memory media; and
- flash media logic configured to interact with different types of the flash memory media…
- wherein the flash abstraction logic invokes the flash media logic to perform memory operations that are potentially performed in different ways depending on the type of the flash memory media, and further wherein the flash memory driver is flash memory medium agnostic…

Claim 1 is rejected under 35 U.S.C. 103(a) as being unpatentable over the combination of Ban, Sinclair, Blumenau, and Hall. The Office argues that Ban teaches:

- one or more computer storage media comprising a flash memory driver that is executable by a computer to interface between a file system and one or more flash memory media, the flash memory driver comprising:
- flash abstraction logic that is invokable by the file system to manage flash memory operations without regard to the type of the one or more flash memory media; and
- flash media logic configured to interact with different types of the flash memory media, wherein the flash media logic is programmable to permit users to match particular medium requirements of a specific manufacturer;

The Office admits that Ban fails to teach that "one of the flash memory operations includes performing wear-leveling operations associated with the flash memory medium by way of circular and continuous advancement of a write point." But the Office argues that Sinclair teaches this element.

The Office also admits that neither Ban nor Sinclair teach "the further limitation of having the flash memory driver residing as a component within the

operating system of the computer system." But the Office argues that Blumenau teaches this element.

The Office also admits that neither Ban, Sinclair, nor Blumenau teach "the flash media logic is programmable to permit users to match particular medium requirements of a specific manufacturer." But the Office argues that Hall teaches this element.

Applicant respectfully submits that Ban does not teach or suggest a flash memory driver comprising flash abstraction logic and flash media logic. Ban describes a system and methodology that is very different from the claimed subject matter. To assist the Office in understanding claim 1, an excerpt from the background section of Appellant's specification is reproduced below:

> **Appellant's Specification, Page 1 (line 17) through Page 2 (line 3)**
>
> Most flash memory manufacturers sell flash memory with ***proprietary controllers***, commonly referred to as ***flash drivers***. Typically, these flash drivers are ***not compatible with flash memories manufactured by other manufacturers***. This reduces flexibility for manufacturers of portable computer devices, because the operating systems and/or file systems deployed by the manufactures are often tied to a particular proprietary flash driver. ***Changing the flash memory medium often requires tedious modification of the file systems, to ensure compatibility with the particular flash driver associated with flash memory medium***. In certain circumstances, some operating as well as file systems are not compatible with certain flash mediums, which may force some manufacturers of portable computer devices to become locked-in, to a certain extent, to a particular flash memory due to a lack of driver compatibility.

Appellant's disclosure addresses problems associated with flash driver-flash medium compatibility (as described above) through the use of a flash memory driver that can be ***programmed*** to adapt to different types of flash memory mediums. In contrast, Ban teaches the use of a ***standardized*** driver that issues generic commands and flash memory mediums equipped with controllers

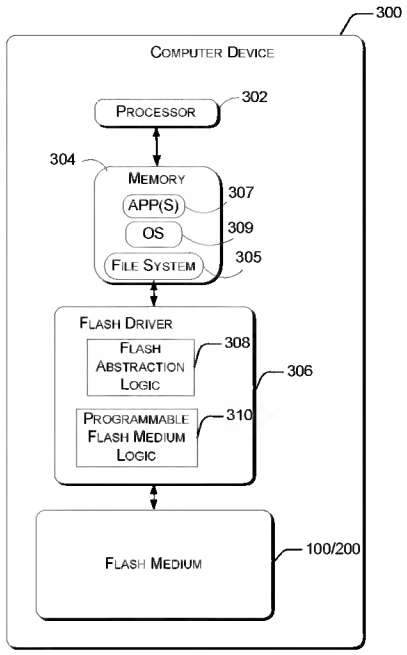
that receive these generic commands. For example, in column 2, lines 35-49, Ban teaches:

> According to the present invention the ***CPU is configured with a standardized driver*** which, according to the present invention, has the ability to interface with any flash chip. The present invention proposes a controller capable of interpreting signals from a standardized driver into commands particular to the flash unit. ***Thus the CPU will produce the commands necessary to perform the flash memory tasks in a uniform, standardized format.*** In this way ***the problem of conforming to the particular requirements of the flash chip is moved from the driver installed on the CPU onto the controller installed on the flash unit.*** The proposed controller is referred to herein as the standardized controller interface.

Accordingly, Ban discloses a system that is markedly different from the subject matter of this claim. As such, Appellant respectfully submits that the Examiner has failed to establish a *prima facie* case of obviousness by improperly equating Ban's "standardized driver" with Applicant's flash memory driver comprising ***flash abstraction logic and flash media logic***.

As an example of subject matter that is within the spirit of the subject matter of this claim, please refer to Fig. 3 of Appellant's disclosure, which is reproduced below for convenience. Fig. 3 depicts a flash memory driver comprising flash abstraction logic and flash media logic.

Fig. 3 of Appellant's Disclosure



As recited in claim 1 the flash memory driver comprises both flash abstraction logic and flash media logic. Claim 1 recites that the flash abstraction logic "is invokable by the file system to manage flash memory operations ***without regard to the one or more flash memory media***, and that the flash media logic is "configured to interact with different types of the flash memory media" and "programmable to permit users to match particular medium requirements of a specific manufacturer". Thus the flash memory driver of claim 1 (which includes both flash abstraction logic and flash media logic) is programmable to ***permit users to match particular medium requirements of a specific manufacturer***.

In contrast, Ban teaches that a CPU is configured with a "***standardized*** driver" (not a programmable driver, as claimed) that is configured to "produce the commands necessary to perform the flash memory tasks in a uniform, standardized format." (See, e.g., Ban, col. 2, lines 41-45.) In other words, Ban teaches that the flash driver produces standardized (or generic) commands which is different than Appellant's driver which "is programmable to permit users to match particular

medium requirements of a specific manufacturer". In order to interpret the generic commands, Ban teaches that each flash chip is equipped with "a controller capable of interpreting signals from a standardized driver into commands particular to the flash unit." (Ban, col. 2, lines 39-41.) Ban teaches that "in this way, the ***problem*** of conforming to the particular requirements of the flash chip is ***moved from the driver*** installed on the CPU ***onto the controller installed on the flash unit***." (Ban, col. 2, lines 43-47, emphasis added.)

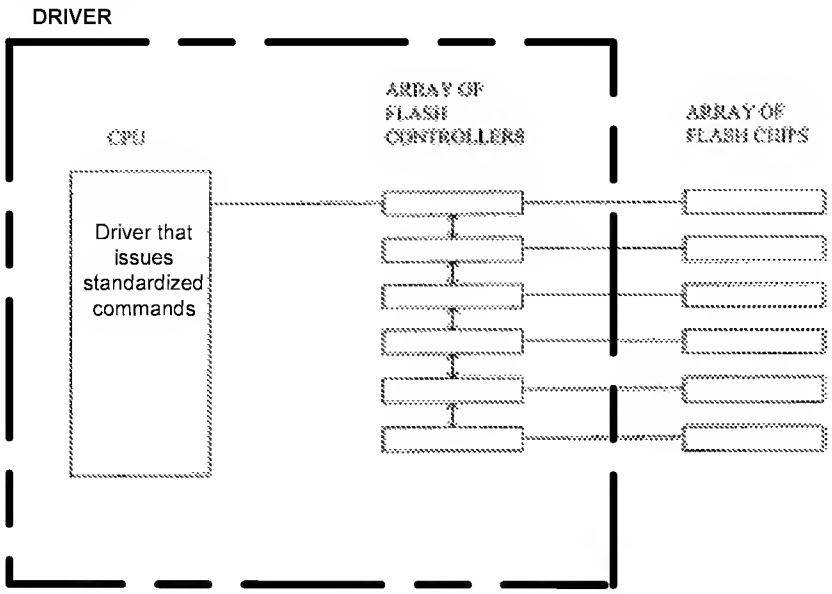
Accordingly, Ban does not teach or in any way suggest a flash memory driver that includes flash media logic which is "programmable to permit users to match particular medium requirements of a specific manufacturer" because Ban specifically teaches that (1) the driver produces standardized commands and (2) that the problem of conforming to the requirements of the flash chip is moved to the flash unit (not the flash driver). To this extent, Sinclair, Blumenau, and Hall fail to add anything of significance.

For at least this reason, the Office has failed to make out a *prima facie* case of obviousness.

Moreover, Ban teaches directly away from a flash driver comprising flash abstraction logic and flash media logic. During an examiner interview conducted on October 23rd, 2007, the Office argued that Ban could be altered to teach the elements of claim 1 by drawing a box around the CPU and the array of flash controllers that is depicted in Fig. 2 of Ban. The Office argued that if this black box were considered "the driver", then Ban would anticipate a flash memory driver comprising flash abstraction logic and flash media logic, as claimed.

In order to illustrate the examiner's argument, Fig. 2 of Ban is reproduced below, with a black box drawn around the CPU and the array of flash controllers (this drawing is also used by the Examiner on page 21 of the Final Office Action dated 11/20/2007):

Fig. 2 of Ban (Altered to Depict a Driver that Includes Flash Controllers)



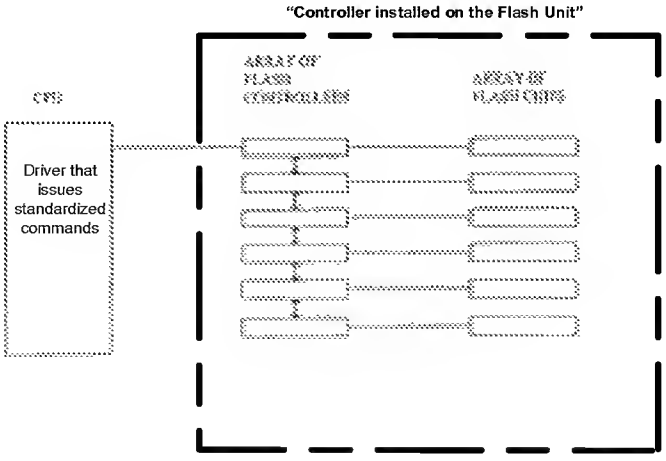
In the drawing above, Fig. 2 of Ban has been ***altered to depict a driver that comprises a standardized driver and an array of flash controllers***. However, Ban teaches directly away from making this modification. Specifically, Ban teaches that "the CPU [which includes the driver] will produce commands necessary to perform the flash memory tasks in a uniform, standardized format." (Ban, col. 2, lines 42-44.) If Ban were to be altered, as illustrated above (and suggested by the Office), Ban's driver would issue commands in a specific format (not a standardized format) because Ban teaches that the controller is configured to produce commands "particular to the flash unit". (See, e.g., Ban, col. 2, lines 41-42.) In fact, in the Final Office Action dated 11/20/2007, the Office agreed with this assertion: "Examiner agreed with Applicant on this point that if the claimed flash memory driver is compared with the black box … drawn around the standardized driver (a part of the CPU) and an array of flash controllers… the driver would issue commands in a specific format (not a standardized format)." (*See* Final Office Action dated 11/20/2007, page 20.)

Furthermore, Ban teaches directly away from making the alteration suggested by the Office because Ban specifically and unequivocally teaches that "in this way, the ***problem*** of conforming to the particular requirements of the flash chip is ***moved from the driver*** installed on the CPU ***onto the controller installed on the flash unit***." (Ban, col. 2, lines 43-47, Emphasis Added.)

If Ban were to be altered, as suggested by the Office, then Ban would operate in a way that is counter to the way that it is disclosed to operate and contrary to the motivation to have it operate that way. Specifically, if Ban were to be altered as suggested by the Office, then the problem of conforming to the particular requirements of the flash chip would be moved to the **flash driver** (and not the controller installed on the ***flash unit***). In other words, if the controllers were moved from the flash unit to the driver, then the flash driver (and not the flash unit) would be responsible for conforming to the particular requirements of the flash unit.

If anything, a black box should be drawn around the array of flash controllers and the array of flash chips, since Ban clearly teaches that the "***controller is installed on the flash unit***." (Ban, col. 2, line 47, emphasis added.) In order to assist the Office, Fig. 2 of Ban has been modified in this manner, below:

Fig. 2 of Ban (Modified to Depict a Controller Installed on the Flash Unit)



By drawing a black box around the flash controllers and flash chips, Fig. 2 depicts that the controller is installed on the flash unit, which is in line with the teachings of Ban ("controller installed on the flash unit"). Modifying Fig. 2 in this manner, which is explicitly supported by Ban, also helps to show that Ban does not contemplate flash controllers residing on the flash driver. Ban simply teaches a standardized driver, and places controllers on the flash unit, instead.

Accordingly, Ban does not teach or in any way suggest a flash driver that comprises flash abstraction logic and flash media logic as recited in this claim. Furthermore, as discussed above, altering Ban by placing the controllers in the flash driver would go directly against the teachings of Ban, and would destroy the very purpose of Ban which is to move the problem of conforming to the particular requirements of the flash chip from the driver to the flash unit. To this extent, Sinclair, Blumenau, and Hall fail to add anything of significance.

For at least this additional reason, the Office has failed to make out a *prima facie* case of obviousness.

In addition, Hall does not disclose or suggest a flash driver comprising flash abstraction logic and flash media logic. On page 13 of the Final Office Action dated 11/20/2007, the Office argues that "even if Ban fails to teach (a) the flash abstraction logic . . . [and] . . . (b) the flash media logic ... ***Hall*** teaches these limitations." (Final Office Action dated 11/20/2007, page 13, emphasis added.) The Office points to column 5, lines 31-48 of Hall as teaching a flash driver comprising flash abstraction logic and flash media logic. For the convenience of the Office, this section of Hall is reproduced below:

> **Hall, Column 5, Lines 31-48**
> While in this embodiment low level code to perform the writing of data into the FLASH memory 22 is included in program ROM within the micro controller 1 and higher level program code is included within the protected area of the FLASH memory this particular partitioning is not essential to the performance of the invention but is of particular convenience in this embodiment. ***The code in the micro controller 1 contains particular sub***

> ***routines which can be selected according to the particular type of FLASH memory (or other non volatile memory) used***. It will be appreciated by those skilled in the art that FLASH memories produced by different manufacturers require different operations to erase and/or write data to them and these sequences are stored for a number of different memories within the microcontroller ROM. Thus the disc drive manufacturer is not confined to a single FLASH memory type and the micro controller does not have to be reprogrammed if a different type of FLASH memory is used.

First, the excerpt above from Hall does not teach or in any way suggest a flash memory driver that comprises flash abstraction logic and flash memory media, as claimed. Specifically, this excerpt from Hall instructs that code in a microcontroller contains subroutines that can be selected according to the particular type of flash memory used. There is no mention of a flash driver that includes flash abstraction logic and flash memory media, as claimed.

Second, the excerpt from Hall teaches directly away from a programmable flash driver. Specifically, the excerpt above states that a "microcontroller" contains different "sub routines that can be ***selected***" for different types of FLASH memories. (Hall, column 5, lines 37-39.) Furthermore, Hall teaches that "the microcontroller ***does not have to be reprogrammed*** if a different type of FLASH memory is used." (Hall, column 5, lines 46-48, emphasis added.) Therefore, Hall teaches directly away from a ***programmable*** flash driver. To this extent, Ban, Sinclair, and Blumenau add nothing of significance.

In view of the discussion above, it is clear that the Office has used ***hindsight reconstruction*** to make out the rejection of this claim. As noted above, Ban teaches directly away from making the alteration suggested by the Office because Ban specifically and unequivocally teaches that "in this way, the ***problem*** of conforming to the particular requirements of the flash chip is ***moved from the driver*** installed on the CPU ***onto the controller installed on the flash unit***." (Ban, col. 2, lines 43-47, Emphasis Added.) Therefore, a person with skill in the art would not think to combine Ban with Blumenau, Sinclair, and Hall in order to

carry out the embodiments taught by Applicant's disclosure. Instead, the Office has clearly used Applicant's disclosure as a template to piece together references in order to make out the rejection of this claim. Absent Applicant's disclosure, one with skill in the art would simply not think of modifying Ban in a way that goes against the very teachings of Ban. Thus, the Office has clearly used hindsight reconstruction to make out the rejection of this claim, which is not permitted by the MPEP. For at least this additional reason, the Office has failed to make out a *prima facie* case of obviousness.

For this and all of the other reasons set forth above, the Office has failed to provide a *prima facie* case of obviousness for the combination of Ban, Sinclair, Blumenau, and Hall.

Claims 3-8

Claims 3-8 depend from Claim 1 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited features that, singly or in combination with those recited in Claim 1, have not been shown to be obvious in the Final Office Action.

Furthermore, the rejection of claims 3 and 4 over Ban, Sinclair, Blumenau, Hall, and ***Martwick***, is not seen to add anything of significance.

Claims 9-11 and 13-15

Claim 9

Appellant respectfully submits that the Office fails to establish a *prima facie* case of obviousness for rejecting Claim 9 in the Final Office Action. The Office has failed to do so first by failing to establish that Ban teaches a flash memory driver comprising flash abstraction logic and flash media logic. Second, the Office improperly argues that Ban could be modified to teach the flash

memory driver, as claimed, because Ban teaches directly away from the modification suggested by the Office. Third, the Office fails to establish that Hall, as used as an alternative to Ban, teaches a flash memory driver comprising flash abstraction logic and flash media logic. Fourth, the Office has used hindsight reconstruction to make out the rejection of this claim. For the reader's convenience, the subject matter of Claim 9 is provided below, after which Appellant submits its arguments for Claim 9.

Claim 9 recites a flash driver embodied on a computer-readable storage medium, comprising:

- flash abstraction logic, interposed between a file system and a flash memory medium, configured to:
  - (a) map a logical sector status from the file system to a physical sector status of the flash memory medium; and
  - (b) maintain memory requirements associated with operating the flash memory medium;
- wherein the flash driver is located remote from the flash memory medium, and wherein the memory requirements include managing wear-leveling operations associated with the flash memory medium by way of circular and continuous advancement of a write pointer, and wherein the flash driver resides as a component within an operating system of a computer;
- flash medium logic that is programmable to permit users to match particular medium requirements of a specific manufacturer.

The Office's argument in support of the rejection of Claim 9 relies on the language and reasoning addressed in Appellant's argument set forth for Claim 1 above. Accordingly, Applicant respectfully submits that Ban does not teach or suggest a flash driver comprising flash abstraction logic and flash medium logic that is programmable to permit users to match particular medium requirements of a specific manufacturer. Furthermore, Ban teaches directly away from a flash

driver comprising flash abstraction logic and flash medium logic that is programmable to permit users to match particular medium requirements of a specific manufacturer, as discussed above with regards to claim 1. Furthermore, Hall does not teach or in any way suggest, and in point of fact teaches directly away from, a flash driver comprising flash abstraction logic and flash medium logic that is programmable to permit users to match particular medium requirements of a specific manufacturer, as discussed above with regards to claim 1. Finally, as discussed above with regards to claim 1, the Office has clearly used impermissible hindsight reconstruction to make out the rejection of this claim.

Accordingly, for any of the reasons set forth above, Appellant submits that the Office has failed to establish a *prima facie* case of obviousness in rejecting Claim 9.

Claims 10, 11, and 13-15

Claims 10, 11, and 13-15 depend from Claim 9 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited features that, singly or in combination with those recited in Claim 9, have not been shown to be obvious in the Final Office Action.

Furthermore, the rejection of claims 13 and 14 over Ban, Sinclair, Blumenau, Hall, and ***Martwick***, is not seen to add anything of significance.

Claims 16-19, 21, and 22

Claim 16

Appellant respectfully submits that the Office fails to establish a *prima facie* case of obviousness for rejecting Claim 16 in the Final Office Action. The

Office has failed to do so first by failing to establish that Ban teaches a flash memory driver comprising flash abstraction logic and flash media logic. Second, the Office improperly argues that Ban could be modified to teach the flash memory driver, as claimed, because Ban teaches directly away from the modification suggested by the Office. Third, the Office fails to establish that Hall, as used as an alternative to Ban, teaches a flash memory driver comprising flash abstraction logic and flash media logic. Fourth, the Office has used hindsight reconstruction to make out the rejection of this claim. For the reader's convenience, the subject matter of Claim 16 is provided below, after which Appellant submits its arguments for Claim 16.

Claim 16 recites a flash driver embodied on a computer-readable storage medium, comprising:

- user programmable flash medium logic, configured to read, write and erase data to and from a flash memory medium, wherein the flash medium logic is programmable to permit users to match particular flash medium requirements of a specific manufacturer; and
- flash abstraction logic, interposed between a file system and flash memory medium to maintain universal requirements for the operation of the flash memory medium;
- wherein the flash memory driver is flash memory medium agnostic, and wherein the universal requirements include managing wear-leveling operations associated with the flash memory medium by way of circular and continuous advancement of a write pointer, and wherein the flash driver is defined as a component within an application.

The Office's argument in support of the rejection of Claim 16 relies on the language and reasoning addressed in Appellant's argument set forth for Claim 1 above. Accordingly, Applicant respectfully submits that Ban does not teach or

suggest a flash driver comprising user programmable flash medium logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer and flash abstraction logic. Furthermore, Ban teaches directly away from a flash driver comprising user programmable flash medium logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer and flash abstraction logic, as discussed above with regards to claim 1. Furthermore, Hall does not teach or in any way suggest, and in point of fact teaches directly away from, a flash driver comprising user programmable flash medium logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer and flash abstraction logic, as discussed above with regards to claim 1. Finally, as discussed above with regards to claim 1, the Office has clearly used impermissible hindsight reconstruction to make out the rejection of this claim.

Accordingly, for any of the reasons set forth above, Appellant submits that the Office has failed to establish a *prima facie* case of obviousness in rejecting Claim 16.

Claims 17-19, 21, and 22

Claims 17-19, 21, and 22 depend from Claim 16 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited features that, singly or in combination with those recited in Claim 16, have not been shown to be obvious in the Final Office Action.

Furthermore, the rejection of claims 19 and 21 over Ban, Sinclair, Blumenau, Hall, and ***Martwick***, is not seen to add anything of significance.

Claims 23-26 and 28-32

Claim 23

Appellant respectfully submits that the Office fails to establish a *prima facie* case of obviousness for rejecting Claim 23 in the Final Office Action. The Office has failed to do so first by failing to establish that Ban teaches a flash memory driver comprising flash abstraction logic and flash media logic. Second, the Office improperly argues that Ban could be modified to teach the flash memory driver, as claimed, because Ban teaches directly away from the modification suggested by the Office. Third, the Office fails to establish that Hall, as used as an alternative to Ban, teaches a flash memory driver comprising flash abstraction logic and flash media logic. Fourth, the Office has used hindsight reconstruction to make out the rejection of this claim. For the reader's convenience, the subject matter of Claim 23 is provided below, after which Appellant submits its arguments for Claim 23.

Claim 23 recites a processing device that uses a flash memory medium for storage of data, comprising:

- a file system, configured to control data storage for the processing device;
- flash media logic, configured to perform physical sector operations to a flash memory medium based on physical sector commands, wherein the flash medium logic comprises a set of programmable entry points that can be implemented by a user to interface with any type of flash memory medium selected, wherein the flash media logic is programmable to permit users to match particular flash medium requirements of a specific manufacturer; and
- flash abstraction logic, configured to maintain flash memory requirements that are necessary to operate the flash memory medium, wherein the flash memory requirements include managing wear-leveling operations associated with the flash memory medium by way of circular and continuous advancement of a write pointer,

wherein the flash media logic and the flash abstraction logic comprise a flash driver.

The Office's argument in support of the rejection of Claim 23 relies on the language and reasoning addressed in Appellant's argument set forth for Claim 1 above. Accordingly, Applicant respectfully submits that Ban does not teach or suggest flash media logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer and flash abstraction logic, wherein the flash media logic and the flash abstraction logic comprise a flash driver. Furthermore, Ban teaches directly away from flash media logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer and flash abstraction logic, wherein the flash media logic and the flash abstraction logic comprise a flash driver, as discussed above with regards to claim 1. Furthermore, Hall does not teach or in any way suggest, and in point of fact teaches directly away from, flash media logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer and flash abstraction logic, wherein the flash media logic and the flash abstraction logic comprise a flash driver, as discussed above with regards to claim 1. Finally, as discussed above with regards to claim 1, the Office has clearly used impermissible hindsight reconstruction to make out the rejection of this claim.

Accordingly, for any of the reasons set forth above, Appellant submits that the Office has failed to establish a *prima facie* case of obviousness in rejecting Claim 23.

Claims 24-26 and 28-32

Claims 24-26 and 28-32 depend from Claim 23 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited features that, singly or in combination with those recited in Claim 23, have not been shown to be obvious in the Final Office Action.

Furthermore, the rejection of claims 26 and 28 over Ban, Sinclair, Blumenau, Hall, and ***Martwick***, is not seen to add anything of significance.

Claims 33-34 and 36-41

Claim 33

Appellant respectfully submits that the Office fails to establish a *prima facie* case of obviousness for rejecting Claim 33 in the Final Office Action. The Office has failed to do so first by failing to establish that Ban teaches a flash memory driver comprising flash abstraction logic and flash media logic. Second, the Office improperly argues that Ban could be modified to teach the flash memory driver, as claimed, because Ban teaches directly away from the modification suggested by the Office. Third, the Office fails to establish that Hall, as used as an alternative to Ban, teaches a flash memory driver comprising flash abstraction logic and flash media logic. Fourth, the Office has used hindsight reconstruction to make out the rejection of this claim. For the reader's convenience, the subject matter of Claim 33 is provided below, after which Appellant submits its arguments for Claim 33.

Claim 33 recites in a processing device that uses a flash memory medium for storage of data, a method for driving the flash memory medium, comprising:

- managing rules associated with operating the flash memory medium in a flash abstraction logic, wherein the flash abstraction logic comprises part of a flash driver; and
- issuing physical sector commands directly to the flash memory medium from a flash medium logic, wherein the flash medium logic is programmable to permit users to match particular flash medium requirements of a specific manufacturer, and wherein the flash medium logic comprises part of the flash driver;
- wherein the method is flash memory medium agnostic, and wherein one of the rules includes managing wear-leveling operations associated with the flash memory medium by way of circular and continuous advancement of a write pointer, and wherein the method is performed by way of a component residing within an operating system of the processing device.

The Office's argument in support of the rejection of Claim 33 relies on the language and reasoning addressed in Appellant's argument set forth for Claim 1 above. Accordingly, Applicant respectfully submits that Ban does not teach or suggest flash abstraction logic, wherein the flash abstraction logic comprises part of a flash driver and flash medium logic, wherein the flash medium logic is programmable to permit users to match particular flash medium requirements of a specific manufacturer and wherein the flash medium logic comprises part of the flash driver. Furthermore, Ban teaches directly away from flash abstraction logic, wherein the flash abstraction logic comprises part of a flash driver and flash medium logic, wherein the flash medium logic is programmable to permit users to match particular flash medium requirements of a specific manufacturer and wherein the flash medium logic comprises part of the flash driver, as discussed above with regards to claim 1. Furthermore, Hall does not teach or in any way suggest, and in point of fact teaches directly away from, flash abstraction logic, wherein the flash abstraction logic comprises part of a flash driver and flash medium logic, wherein the flash medium logic is programmable to permit users to

match particular flash medium requirements of a specific manufacturer and wherein the flash medium logic comprises part of the flash driver, as discussed above with regards to claim 1. Finally, as discussed above with regards to claim 1, the Office has clearly used impermissible hindsight reconstruction to make out the rejection of this claim.

Accordingly, for any of the reasons set forth above, Appellant submits that the Office has failed to establish a *prima facie* case of obviousness in rejecting Claim 33.

Claims 34 and 36-41

Claims 34 and 36-41 depend from Claim 33 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited features that, singly or in combination with those recited in Claim 33, have not been shown to be obvious in the Final Office Action.

Furthermore, the rejection of claims 34 and 36 over Ban, Sinclair, Blumenau, Hall, and ***Martwick***, is not seen to add anything of significance.

Claim 42

Appellant respectfully submits that the Office fails to establish a *prima facie* case of obviousness for rejecting Claim 42 in the Final Office Action. The Office has failed to do so first by failing to establish that Ban teaches a flash memory driver comprising flash abstraction logic and flash media logic. Second, the Office improperly argues that Ban could be modified to teach the flash memory driver, as claimed, because Ban teaches directly away from the modification suggested by the Office. Third, the Office fails to establish that Hall,

as used as an alternative to Ban, teaches a flash memory driver comprising flash abstraction logic and flash media logic. Fourth, the Office has used hindsight reconstruction to make out the rejection of this claim. For the reader's convenience, the subject matter of Claim 42 is provided below, after which Appellant submits its arguments for Claim 42.

Claim 42 recites a computer storage media for a flash driver, comprising computer-executable instructions that, when executed, direct the flash driver to provide an interface between a file system, selected from one of a plurality of different file systems, and a flash memory medium, selected from one of a plurality of different flash memory media, wherein the flash driver is located as a component within an operating system and is remote from the flash memory medium, and wherein wear-leveling of the flash memory medium is performed by way of circular and continuous advancement of a write pointer, and wherein the flash driver comprises programmable flash medium logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer.

The Office's argument in support of the rejection of Claim 42 relies on the language and reasoning addressed in Appellant's argument set forth for Claim 1 above. Accordingly, Applicant respectfully submits that Ban does not teach or suggest a flash driver comprising programmable flash medium logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer. Furthermore, Ban teaches directly away from a flash driver comprising programmable flash medium logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer, as

discussed above with regards to claim 1. Furthermore, Hall does not teach or in any way suggest, and in point of fact teaches directly away from, a flash driver comprising programmable flash medium logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer, as discussed above with regards to claim 1. Finally, as discussed above with regards to claim 1, the Office has clearly used impermissible hindsight reconstruction to make out the rejection of this claim.

Accordingly, for any of the reasons set forth above, Appellant submits that the Office has failed to establish a *prima facie* case of obviousness in rejecting Claim 42.

Claim 43

Appellant respectfully submits that the Office fails to establish a *prima facie* case of obviousness for rejecting Claim 43 in the Final Office Action. The Office has failed to do so first by failing to establish that Ban teaches a flash memory driver comprising flash abstraction logic and flash media logic. Second, the Office improperly argues that Ban could be modified to teach the flash memory driver, as claimed, because Ban teaches directly away from the modification suggested by the Office. Third, the Office fails to establish that Hall, as used as an alternative to Ban, teaches a flash memory driver comprising flash abstraction logic and flash media logic. Fourth, the Office has used hindsight reconstruction to make out the rejection of this claim. For the reader's convenience, the subject matter of Claim 43 is provided below, after which Appellant submits its arguments for Claim 43.

Claim 43 recites a computer storage media for a flash driver, comprising computer-executable instructions that, when executed, direct the flash driver to:

- provide an interface between a file system, selected from one of a plurality of different files systems, and a flash memory medium, selected from one of a plurality of different flash memory media; and
- manage a set of characteristics that are common to the plurality of different flash memory media at a flash abstraction logic;
- wherein the flash driver is flash memory medium agnostic, and wherein wear-leveling of the flash memory medium is performed by way of circular and continuous advancement of a write pointer, and wherein the flash driver resides as a component within an operating system, wherein the instructions provide programmable flash medium logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer, wherein the flash abstraction logic and the flash medium logic comprise the flash driver.

The Office's argument in support of the rejection of Claim 43 relies on the language and reasoning addressed in Appellant's argument set forth for Claim 1 above. Accordingly, Applicant respectfully submits that Ban does not teach or suggest a flash driver comprising flash abstraction logic and programmable flash medium logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer, wherein the flash abstraction logic and the flash medium logic comprise the flash driver. Furthermore, Ban teaches directly away from a flash driver comprising flash abstraction logic and programmable flash medium logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer, wherein the flash abstraction logic and the flash medium logic comprise the flash driver, as discussed above with regards to claim 1. Furthermore, Hall does not teach or in

any way suggest, and in point of fact teaches directly away from, a flash driver comprising flash abstraction logic and programmable flash medium logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer, wherein the flash abstraction logic and the flash medium logic comprise the flash driver, as discussed above with regards to claim 1. Finally, as discussed above with regards to claim 1, the Office has clearly used impermissible hindsight reconstruction to make out the rejection of this claim.

Accordingly, for any of the reasons set forth above, Appellant submits that the Office has failed to establish a *prima facie* case of obviousness in rejecting Claim 43.

Claim 44

Appellant respectfully submits that the Office fails to establish a *prima facie* case of obviousness for rejecting Claim 44 in the Final Office Action. The Office has failed to do so first by failing to establish that Ban teaches a flash memory driver comprising flash abstraction logic and flash media logic. Second, the Office improperly argues that Ban could be modified to teach the flash memory driver, as claimed, because Ban teaches directly away from the modification suggested by the Office. Third, the Office fails to establish that Hall, as used as an alternative to Ban, teaches a flash memory driver comprising flash abstraction logic and flash media logic. Fourth, the Office has used hindsight reconstruction to make out the rejection of this claim. For the reader's convenience, the subject matter of Claim 44 is provided below, after which Appellant submits its arguments for Claim 44.

Claim 44 recites a computer storage media for a flash driver, comprising computer-executable instructions that, when executed, direct the flash driver to:

- provide an interface between a file system, selected from one of a plurality of different files systems, and a flash memory medium, selected from one of a plurality of different flash memory media;
- manage a set of characteristics that are common to the plurality of different flash memory media at a flash abstraction logic; and
- provide programmable entry points that can be optionally selected by a user to interface with the type of flash memory medium selected;
- wherein the flash driver is located as a component within an operating system and is remote from the flash memory medium and the flash driver is flash memory medium agnostic, and wherein wear-leveling of a flash memory medium is performed by way of circular and continuous advancement of a write pointer, wherein the instructions provide flash medium logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer, wherein the flash abstraction logic and the flash medium logic comprise the flash driver.

The Office's argument in support of the rejection of Claim 44 relies on the language and reasoning addressed in Appellant's argument set forth for Claim 1 above. Accordingly, Applicant respectfully submits that Ban does not teach or suggest a flash driver comprising flash abstraction logic and programmable flash medium logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer, wherein the flash abstraction logic and the flash medium logic comprise the flash driver. Furthermore, Ban teaches directly away from a flash driver comprising flash abstraction logic and programmable flash medium logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer, wherein the flash abstraction logic and the flash medium logic comprise the flash driver, as

discussed above with regards to claim 1. Furthermore, Hall does not teach or in any way suggest, and in point of fact teaches directly away from, a flash driver comprising flash abstraction logic and programmable flash medium logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer, wherein the flash abstraction logic and the flash medium logic comprise the flash driver, as discussed above with regards to claim 1. Finally, as discussed above with regards to claim 1, the Office has clearly used impermissible hindsight reconstruction to make out the rejection of this claim.

For any of the reasons set forth above, Appellant submits that the Office has failed to establish a *prima facie* case of obviousness in rejecting Claim 44.

Conclusion

Appellant respectfully submits that all of the Examiner's rejections have been traversed. As such, Appellant respectfully submits that all of the claims are in condition for allowance.

Respectfully Submitted,

Dated: April 21, 2008

By: /Mark F. Niemann/

Mark F. Niemann
Sadler, Breen, Morasch & Colby, P.S.
Reg. No. 61,817
(509) 755-7259

(8) Appendix of Appealed Claims

1. (Previously Presented) One or more computer storage media comprising a flash memory driver that is executable by a computer to interface between a file system and one or more flash memory media, the flash memory driver comprising:

flash abstraction logic that is invokable by the file system to manage flash memory operations without regard to the type of the one or more flash memory media; and

flash media logic configured to interact with different types of the flash memory media, wherein the flash media logic is programmable to permit users to match particular medium requirements of a specific manufacturer;

wherein the flash abstraction logic invokes the flash media logic to perform memory operations that are potentially performed in different ways depending on the type of the flash memory media, and further wherein the flash memory driver is flash memory medium agnostic, and wherein one of the flash memory operations includes performing wear-leveling operations associated with the flash memory medium by way of circular and continuous advancement of a write pointer, and wherein the flash memory driver resides as a component within an operating system of the computer.

2. (Canceled)

3. (Original) The flash memory driver as recited in Claim 1, wherein one of the flash memory operations includes maintaining data integrity of the flash memory medium.

4. (Original) The flash memory driver as recited in Claim 1, wherein one of the flash memory operations includes handling recovery of data associated with the flash memory medium after a power-failure.

5. (Original) The flash memory driver as recited in Claim 1, wherein one of the flash memory operations includes mapping status information associated with physical sectors of the flash memory medium for use by the file system.

6. (Previously Presented) The flash memory driver as recited in Claim 1, wherein the flash medium logic is further configured to translate commands received from the file system to physical sector commands for issuance to the flash memory media.

7. (Previously Presented) The flash memory driver as recited in Claim 1, wherein the flash medium logic is user programmable to read, write and erase data to and from the flash memory media.

8. (Previously Presented) The flash memory driver as recited in Claim 1, wherein the flash media logic is configured to perform error code correction associated with the flash memory media.

9. (Previously Presented) A flash driver embodied on a computer-readable storage medium, comprising:

flash abstraction logic, interposed between a file system and a flash memory medium, configured to:

(a) map a logical sector status from the file system to a physical sector status of the flash memory medium; and

(b) maintain memory requirements associated with operating the flash memory medium;

wherein the flash driver is located remote from the flash memory medium, and wherein the memory requirements include managing wear-leveling operations associated with the flash memory medium by way of circular and continuous advancement of a write pointer, and wherein the flash driver resides as a component within an operating system of a computer;

flash medium logic that is programmable to permit users to match particular medium requirements of a specific manufacturer.

10. (Previously Presented) The flash driver as recited in Claim 9, wherein the programmable flash medium logic is configured to read, write and erase data to and from the flash memory medium.

11. (Previously Presented) The flash driver as recited in Claim 9, wherein the programmable flash medium logic is configured to receive and translate specific operational commands from the file system associated with reading and writing data to the flash memory medium.

12. (Canceled)

13. (Original) The flash driver as recited in Claim 9, wherein the memory requirements include maintaining data integrity of the flash memory medium.

14. (Original) The flash driver as recited in Claim 9, wherein the memory requirements include handling recovery of data associated with flash memory medium after a power-failure.

15. (Previously Presented) The flash driver as recited in Claim 9, wherein the flash medium logic is programmably configurable by a user to perform error code correction associated with the flash memory medium.

16. (Previously Presented) A flash driver embodied on a computer-readable storage medium, comprising:

user programmable flash medium logic, configured to read, write and erase data to and from a flash memory medium, wherein the flash medium logic is programmable to permit users to match particular flash medium requirements of a specific manufacturer; and

flash abstraction logic, interposed between a file system and flash memory medium to maintain universal requirements for the operation of the flash memory medium;

wherein the flash memory driver is flash memory medium agnostic, and wherein the universal requirements include managing wear-leveling operations associated with the flash memory medium by way of circular and continuous advancement of a write pointer, and wherein the flash driver is defined as a component within an application.

17. (Original) The flash driver as recited in Claim 16, wherein the flash abstraction logic passes specific commands associated with certain types of flash memory media directly to the flash medium logic for translation and execution.

18. (Original) The flash driver as recited in Claim 16, wherein the flash abstraction logic is an interface between the flash medium logic and the file system.

19. (Original) The flash driver as recited in Claim 16, wherein the universal requirements include maintaining data integrity of the flash memory medium.

20. (Canceled)

21. (Original) The flash driver as recited in Claim 16, wherein the universal requirements include handling recovery after a power-failure.

22. (Original) The flash driver as recited in Claim 16, wherein the flash medium logic comprises a set of programmable entry points that can be implemented by a user to interface with the type of flash memory medium selected.

23. (Previously Presented) A processing device that uses a flash memory medium for storage of data, comprising:

a file system, configured to control data storage for the processing device;

flash media logic, configured to perform physical sector operations to a flash memory medium based on physical sector commands, wherein the flash medium logic comprises a set of programmable entry points that can be implemented by a user to interface with any type of flash memory medium selected, wherein the flash media logic is programmable to permit users to match particular flash medium requirements of a specific manufacturer; and

flash abstraction logic, configured to maintain flash memory requirements that are necessary to operate the flash memory medium, wherein the flash memory requirements include managing wear-leveling operations associated with the flash memory medium by way of circular and continuous advancement of a write pointer, wherein the flash media logic and the flash abstraction logic comprise a flash driver.

24. (Original) The processing device as recited in Claim 23, wherein the flash abstraction logic passes physical logic commands associated with certain types of flash memory medium directly to the flash memory medium logic for translation and execution.

25. (Original) The processing device as recited in Claim 23, wherein the flash abstraction logic is an interface between the flash medium logic and the file system.

26. (Original) The processing device as recited in Claim 23, wherein the flash memory requirements include maintaining data integrity of the flash memory medium.

27. (Canceled)

28. (Original) The processing device as recited in Claim 23, wherein the flash memory requirements include handling recovery after a power-failure.

29. (Original) The processing device as recited in Claim 23, wherein the requirements are common to a plurality of different flash memory media.

30. (Original) The processing device as recited in Claim 23, wherein the flash medium logic comprises a set of programmable entry points that can be implemented by a user to perform error code correction with the type of flash memory medium used in the processing device.

31. (Original) The processing device as recited in Claim 23, whereby the flash medium logic relieves the flash abstraction logic from performing translation of the physical sector commands received from the file system.

32. (Original) The processing device as recited in Claim 23, wherein the physical sector operations include read, write and error code correction commands associated with the flash memory medium.

33. (Previously Presented) In a processing device that uses a flash memory medium for storage of data, a method for driving the flash memory medium, comprising:

managing rules associated with operating the flash memory medium in a flash abstraction logic, wherein the flash abstraction logic comprises part of a flash driver; and

issuing physical sector commands directly to the flash memory medium from a flash medium logic, wherein the flash medium logic is programmable to permit users to match particular flash medium requirements of a specific manufacturer, and wherein the flash medium logic comprises part of the flash driver;

wherein the method is flash memory medium agnostic, and wherein one of the rules includes managing wear-leveling operations associated with the flash memory medium by way of circular and continuous advancement of a write pointer, and wherein the method is performed by way of a component residing within an operating system of the processing device.

34. (Original) The method as recited in Claim 33, wherein one of the rules includes maintaining data integrity of the flash memory medium.

35. (Canceled)

36. (Original) The method as recited in Claim 33, wherein one of the rules includes handling recovery of the media after a power-failure.

37. (Original) The method as recited in Claim 33, wherein issuing physical sector commands directly to the flash memory medium comprises receiving read and write commands from a file system and translating them into the physical sector commands.

38. (Original) The method as recited in Claim 33, further comprising issuing a set of programmable entry points that can be implemented by a user to perform error code correction with the type of flash memory medium used in the processing device.

39. (Original) The method as recited in Claim 33, further comprising issuing a set of programmable entry points that can be optionally selected by a user to interface with the type of flash memory medium used in the processing device.

40. (Original) The method as recited in Claim 33, further comprising receiving read and write commands from a file system.

41. (Previously Presented) One or more computer-readable storage media comprising computer-executable instructions that, when executed, perform the method as recited in claim 33.

42. (Previously Presented) A computer storage media for a flash driver, comprising computer-executable instructions that, when executed, direct the flash driver to provide an interface between a file system, selected from one of a

plurality of different file systems, and a flash memory medium, selected from one of a plurality of different flash memory media, wherein the flash driver is located as a component within an operating system and is remote from the flash memory medium, and wherein wear-leveling of the flash memory medium is performed by way of circular and continuous advancement of a write pointer, and wherein the flash driver comprises programmable flash medium logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer.

43. (Previously Presented) A computer storage media for a flash driver, comprising computer-executable instructions that, when executed, direct the flash driver to:

provide an interface between a file system, selected from one of a plurality of different files systems, and a flash memory medium, selected from one of a plurality of different flash memory media; and

manage a set of characteristics that are common to the plurality of different flash memory media at a flash abstraction logic;

wherein the flash driver is flash memory medium agnostic, and wherein wear-leveling of the flash memory medium is performed by way of circular and continuous advancement of a write pointer, and wherein the flash driver resides as a component within an operating system, wherein the instructions provide programmable flash medium logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer, wherein the flash abstraction logic and the flash medium logic comprise the flash driver.

44. (Previously Presented) A computer storage media for a flash driver, comprising computer-executable instructions that, when executed, direct the flash driver to:

provide an interface between a file system, selected from one of a plurality of different files systems, and a flash memory medium, selected from one of a plurality of different flash memory media;

manage a set of characteristics that are common to the plurality of different flash memory media at a flash abstraction logic; and

provide programmable entry points that can be optionally selected by a user to interface with the type of flash memory medium selected;

wherein the flash driver is located as a component within an operating system and is remote from the flash memory medium and the flash driver is flash memory medium agnostic, and wherein wear-leveling of a flash memory medium is performed by way of circular and continuous advancement of a write pointer, wherein the instructions provide flash medium logic that is programmable to permit users to match particular flash medium requirements of a specific manufacturer, wherein the flash abstraction logic and the flash medium logic comprise the flash driver.

45.-52. (Canceled)

(9) Evidence Appendix: None

(10) Related proceedings Appendix: None